WHAT IS CLAIMED IS:

1.   A database cluster which avoids client failure by connecting to multiple nodes of the cluster, the database cluster comprising:

    a first computing system including:

        a primary connection manager which forms a client connection with and receives transactions from at least one client, and

        a primary database management system (DBMS) which communicates with the primary connection manager to receive the transactions and executes the transactions on data stored in one or more data files; and

    a second computing system including:

        a secondary connection manager, and

        a secondary DBMS which communicates with the secondary connection manager and can access data stored in the one or more data files,

    wherein when the second connection manager determines that a predetermined condition is met, the second connection manager receives data from the client connection, replays incomplete portions of open transactions on the data through the secondary DBMS, and begins to receive additional transactions from the at least one client to be executed against the one or more data files.

2.   The highly available database cluster of Claim 1, wherein the predetermined condition comprises a failure of the first computing system.

3.   The highly available database cluster of Claim 1, wherein the predetermined condition comprises a failure of the primary DBMS.

4.   The highly available database cluster of Claim 1, wherein the predetermined condition comprises an unbalanced workload between the first and second computing systems.

5.   The highly available database cluster of Claim 1, wherein the primary connection manager and the secondary connection manager communicate with one another.

6.   The highly available database cluster of Claim 5, wherein the primary connection manager transmits copies to the secondary connection manager of data packets which include the transactions and responses or acknowledgements to the transactions.

7.   The highly available database cluster of Claim 5, wherein the primary connection manager and the secondary connection manager exchange statistics in order to monitor the client connection.

8.   The highly available database cluster of Claim 7, wherein the statistics include the number of clients connected to the primary connection manager.

9.   The highly available database cluster of Claim 7, wherein the statistics include the number of clients the secondary connection manager can see connected to the primary connection manager.

10.   The highly available database cluster of Claim 7, wherein the statistics include whether the secondary connection manager can communicate with the primary connection manager.

11.   A primary and at least one secondary connection manager of a database cluster, which manage a connection between at least one client and two or more database management systems (DBMSs), wherein the primary and at least one secondary connection manager can move the connection from the primary connection manager to the at least one secondary connection manager while providing protocols for the connection native to the two or more DBMSs, the primary and secondary connection manager comprising:

a first memory;

a primary connection configured to form a connection with a client and to place statements from transactions from the client into the first memory;

a primary protocol shadow configured to retrieve the statements and forward the statements to a primary DBMS;

a secondary memory;

a secondary connection configured to receive transactions from the connection with the client when one or more predetermined conditions are met and to place new statements from the transactions from the client into the second memory;

at least one process configured to replay any incomplete statements of open transactions; and

a secondary protocol shadow configured to connect to the at least one process until the incomplete statements are forwarded to a secondary DBMS and then to connect to the secondary memory to retrieve the new statements and forward the new statements to the secondary DBMS.

12.   The primary and at least one secondary connection manager of Claim 11, wherein the protocol native to the two or more DBMSs comprises SQL*Net.

13. The primary and at least one secondary connection manager of Claim 11, wherein the at least one process further comprises:

an import process configured to retrieve the statements from the primary connection and store those statements associated with open transactions; and

a replay process configured to access the stored statements and to forward the stored statements to the secondary protocol shadow.

14. The primary and at least one secondary connection manager of Claim 11, wherein the secondary protocol shadow is configured to access a log file of the primary DBMS to ensure against replaying of statements of closed transactions.

15. The primary and at least one secondary connection manager of Claim 11, wherein the primary and secondary connections communicate with one another.

16. The primary and at least one secondary connection manager of Claim 15, wherein the primary connection and the secondary connection exchange statistics in order to monitor the connection.

17. A method of providing native protocol access and transparent fail-over to a client connection thereby avoiding a client failure when a primary host fails, the method comprising:

rerouting a client connection between a first host and a client to a second host;

replaying at least one statement from open transactions, wherein the at least one statement includes a statement received but not committed by the first host when the client connection was moved from the first host; and

establishing communication between the second host and the client over the client connection.

18. The method of Claim 17, further comprising sending keepalive messages to the client in order to keep the client from dropping the client connection.

19. The method of Claim 17, wherein the replaying at least one statement further comprising removing leftover statements of closed transactions.

20. A method of providing transparent fail-over to a client connection thereby avoiding a client failure when a primary database management system DBMS fails, the method comprising:

monitoring statistics of a client connection between a first DBMS and a client;

determining from the statistics a need to move the client connection to a second DBMS while keeping the client connection alive from a perspective of the client;

rerouting the client connection to the second DBMS;

replaying any statements from open transactions rolled back when the client connection was moved from the first DBMS; and

establishing communication between the second DBMS and the client over the client connection.

21. A data processing system which provides transparent fail-over to a client connection, thereby avoiding a client failure when a primary host fails, the data processing system comprising:

a first host configured to accept a client connection from a client;

a connection manager which reroutes the client connection to a second host without recognition by the client; and

a replay process which forwards to the second host at least one incomplete statement from open transactions when the client connection was moved from the first host, wherein the connection manager establishes communication between the second host and the client over the client connection.

22.   The data processing system of Claim 21, wherein the client communication comprises a protocol native to the primary host.

23.   The data processing system of Claim 22, wherein the protocol comprises SQL*Net.